IV International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2011
M. Papadrakakis, E. Oñate and B. Schrefler (Eds)

AN ITERATION-FREE, PARTITIONED METHOD FOR SOLVING COUPLED PROBLEMS

JOACHIM RANG*

*Institute of Scientific Computing
TU Braunschweig
Hans-Sommer Str. 65, 38106 Braunschweig, Germany
e-mail: j.rang@tu-bs, web page: http://www.wire.tu-bs.de

Key words: Coupled problems, partitioned methods, linear–implicit time integration, Rosenbrock–Wanner methods

Abstract. Coupled problems consist of two or more problems which in most cases describe different physical phenomena. An example of such a problem is the interaction of fluid and structure. Usually, the most accurate way to solve coupled problems is the monolithical approach. But often, due to different reasons, a partitioned method is used, where the subproblems are solved with different software packages and there may be different discretisation methods. One reason for partitioning a coupled problem is that existing codes and the best discretisation schemes can be used. In this note we introduce an iteration-free, partitioned method which is based on a linear-implicit time integration method.

1 INTRODUCTION

Coupled problems appear in different research areas. One common example is the interaction of structure and fluid [DR08], e.g. the numerical simulation of offshore wind turbines, see [MM04], or of biomechanical processes. Coupled problems consist of two or more different physical problems which are in general space and time dependent. The discretisation in space leads to a high dimensional system of ordinary differential equations (ODEs). The computation of the numerical solution needs the simultaneous solution of the strong coupled equations of each problem. But often for each subproblem different discretisation schemes are used. In the case of fluid-structure interaction the fluid is discretised with Finite Volumes and the structure with Finite Elements. For building a monolithic solver [RB00], it is often difficult to find a free available software system which processes different discretisation methods for different problem classes.

This is one reason to use a modular approach and partitioned methods [RB00, FP80, MW01, PFL95, MS02, MNS06], i.e. the subproblems are solved by different codes which

communicate with each other. The communication between the solvers can be realised with the help of the Component Template Library (CTL), i.e. the solvers are transformed into software components and are controlled from outside with a central unit. In [RSM09] the CTL is used to solve FSI problems.

For the time discretisation of parabolic differential equations, the heat equation or the incompressible Navier–Stokes equations, often implicit methods are used [GS00, JR10] to obtain a stable numerical solution. To use an implicit time integration method for a coupled problem leads to difficulties in solving the final non-linear system since each solver processes only a part of the system. Therefore iterative methods as the staggered scheme, the Block-Gauß-Seidel- or the Block–Newton scheme are used to solve this non-linear system.

In this paper we introduce two classes of time stepping schemes. First the diagonally implicit Runge–Kutta methods (DIRK–methods). We formulate the Block-Gauß-Seidel- and the Block–Newton scheme. The second class are the linear implicit Runge–Kutta methods, the so-called Rosenbrock–Wanner methods. This class of methods needs only the solution of a linear system. It is possible to formulate the Block–Gauß–Seidel method in such a way that we get an iteration-free partitioned method, and the Block–Newton method reduces to a Block–Gauß method, i.e. only one iteration step is needed.

The paper is structured as follows: First we give a short introduction into the time discretisation schemes. Then the Block–Gauß–Seidel and the Block–Newton methods for both discretisation schemes are formulated. In chapter 4 we present a numerical result.

2 TIME DISCRETISATION

In this note we are considering strongly coupled problems of ODEs which are given by

$$M_1\dot{\mathbf{u}} = \mathbf{f}(t,\mathbf{u},\mathbf{v}), \quad \mathbf{u}(0) = \mathbf{u}_0 \tag{1}$$

$$M_2\dot{\mathbf{v}} = \mathbf{g}(t,\mathbf{u},\mathbf{v}), \quad \mathbf{v}(0) = \mathbf{v}_0, \tag{2}$$

where the matrices $M_1 \in \mathbb{R}^{n_1,n_1}$ and $M_2 \in \mathbb{R}^{n_2,n_2}$ are regular. Problems which can be formulated in the form (1)–(2) arise in the simulation of mechanical problems, in the case of semi-discretised Dirichlet-Neumann-problems and in the simulation of FSI problems (see [MS03]).

In practical applications the systems (1) and (2) are stiff, i. e. explicit time disscretisations schemes need arbitrarily small time steps to compute a stable numerical solution. Therefore we consider linear-implicit and diagonally implicit Runge–Kutta methods which have no steplength restriction to produce a stable numerical solution.

2.1 Diagonally implicit Runge–Kutta methods

Application to ODEs. First we consider an implicite ODE of the form

$$M\dot{\mathbf{u}} = \mathbf{F}(t,\mathbf{u}), \quad \mathbf{u}(0) = \mathbf{u}_0, \tag{3}$$

where M is a regular matrix. A Runge–Kutta method (RK–method) with s internal stages, [HW96, SW92], is a one–step–method for solving (3) of the form

$$M\mathbf{k}_i = \mathbf{F}\left(t_m + c_i\tau_m, \mathbf{U}_i\right), \quad \mathbf{U}_i = \mathbf{u}_m + \tau_m \sum_{j=1}^{s} a_{ij}\mathbf{k}_j, \quad i = 1, \ldots, s, \tag{4}$$

$$\mathbf{u}_{m+1} = \mathbf{u}_m + \tau_m \sum_{i=1}^{s} b_i\mathbf{k}_i. \tag{5}$$

The coefficients of an RK–method are usually represented with the help of a Butcher–table,

$$\begin{array}{c|ccc} c_1 & a_{11} & \ldots & a_{1s} \\ c_2 & a_{21} & \ldots & a_{2s} \\ \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \ldots & a_{ss} \\ \hline & b_1 & \ldots & b_s \end{array} = \begin{array}{c|c} \mathbf{c} & A \\ \hline & \mathbf{b}^\top \end{array}.$$

The vector $\mathbf{c}$ includes the grid points of the time discretisation and $\mathbf{b}$ is a vector with weights. The coefficients a_{ij}, b_i and c_i should be chosen in such a way that some order conditions are satisfied to obtain a sufficient consistency order.

In this paper, the coefficients of the RK–method (4)–(5) satisfy $a_{ij} = 0$ for $i < j$, $i, j \in \{1, \ldots, s\}$ and $a_{ii} \neq 0$ for $i \in \{2, \ldots, s\}$. RK–methods satisfying these conditions are called diagonal–implicit RK–methods (DIRK–methods). These methods are discussed in several papers and books, e.g. in [SW92, HW96]. Applications to fluid problems can be found in [JGR06, JR10] and to structural problems in [HH10].

Application to strongly coupled problems. Next we apply the RK–method (4)–(5) on our strongly coupled problem (1)–(2). Then the method reads as

$$M_1\mathbf{k}_i = \mathbf{f}\left(t_m + c_i\tau_m, \mathbf{U}_i, \mathbf{V}_i\right), \quad \mathbf{U}_i = \mathbf{u}_m + \tau_m \sum_{j=1}^{s} a_{ij}\mathbf{k}_j, \quad i = 1, \ldots, s, \tag{6}$$

$$M_2\mathbf{l}_i = \mathbf{g}\left(t_m + c_i\tau_m, \mathbf{U}_i, \mathbf{V}_i\right), \quad \mathbf{V}_i = \mathbf{v}_m + \tau_m \sum_{j=1}^{s} a_{ij}\mathbf{l}_j, \quad i = 1, \ldots, s, \tag{7}$$

$$\mathbf{u}_{m+1} = \mathbf{u}_m + \tau_m \sum_{i=1}^{s} b_i\mathbf{k}_i, \quad \mathbf{v}_{m+1} = \mathbf{v}_m + \tau_m \sum_{i=1}^{s} b_i\mathbf{l}_i. \tag{8}$$

In each timestep s non-linear systems have to be solved. One possibility for the solution of these systems is the simplified Newton method [SW92, HW96] which reads for the equation $\mathbf{F}(\mathbf{x}) = 0$ as follows

$$\mathbf{x}^{(\nu+1)} = \mathbf{x}^{(\nu)} - (\partial_{\mathbf{x}}\mathbf{F}(\mathbf{x}^{(0)}))^{-1}\mathbf{F}(\mathbf{x}^{(\nu)}). \tag{9}$$

Since inverting a matrix is very expensive, we multiply by $\partial_{\mathbf{x}}\mathbf{F}(\mathbf{x}^{(0)})$ and get the linear system

$$(\partial_{\mathbf{x}}\mathbf{F}(\mathbf{x}^{(0)}))(\mathbf{x}^{(\nu+1)} - \mathbf{x}^{(\nu)}) = -\mathbf{F}(\mathbf{x}^{(\nu)}).$$

In the case $\gamma := a_{ii}$ for $i = 1, \ldots, s$, the system-matrix on the left-hand side in (6)–(7) does not change during a timestep. Therefore we can make one LU-decomposition in each time-step and then solve all non-linear systems by forward and backward substitutions which reduce the cost for the linear algebra. In our case the linear systems reads as

$$\begin{pmatrix} M_1 - \tau a_{ii}\partial_{\mathbf{u}}\mathbf{f}_m & -\tau a_{ii}\partial_{\mathbf{v}}\mathbf{f}_m \\ -\tau a_{ii}\partial_{\mathbf{u}}\mathbf{g}_m & M_2 - \tau a_{ii}\partial_{\mathbf{v}}\mathbf{g}_m \end{pmatrix} \left[\begin{pmatrix} \mathbf{k}_i^{(\nu+1)} \\ \mathbf{l}_i^{(\nu+1)} \end{pmatrix} - \begin{pmatrix} \mathbf{k}_i^{(\nu)} \\ \mathbf{l}_i^{(\nu)} \end{pmatrix} \right]$$
$$= \begin{pmatrix} M_1 & 0 \\ 0 & M_2 \end{pmatrix} \begin{pmatrix} \mathbf{k}_i^{(\nu)} \\ \mathbf{l}_i^{(\nu)} \end{pmatrix} - \begin{pmatrix} \mathbf{f}\left(t_m + c_i\tau_m, \mathbf{U}_i^{(\nu)}, \mathbf{V}_i^{(\nu)}\right) \\ \mathbf{g}\left(t_m + c_i\tau_m, \mathbf{U}_i^{(\nu)}, \mathbf{V}_i^{(\nu)}\right) \end{pmatrix}, \quad (10)$$

where $\nu > 0$, $\partial_{\mathbf{u}}\mathbf{f}_m := \partial_{\mathbf{u}}\mathbf{f}(t_m, \mathbf{u}, \mathbf{v})$, and

$$\mathbf{U}_i^{(\nu)} := \mathbf{u}_m + \tau_m \sum_{j=1}^{i-1} a_{ij}\mathbf{k}_j + \tau_m a_{ii}\mathbf{k}_i^{(\nu)}, \qquad i = 1, \ldots, s$$

Adaptive time step control. RK–methods have the advantage that they allow an easy implementation of an adaptive time steplength control. Consider a RK–method of order $p \geq 2$. An adaptive time step control employs a second RK–method which has the coefficients a_{ij}, $\hat{b}_i$ and c_i, $i, j = 1, \ldots, s$, and order $p - 1$. The solution of the second method at t_{m+1} is given by

$$\hat{\mathbf{u}}_{m+1} = \mathbf{u}_m + \sum_{i=1}^{s} \hat{b}_i\mathbf{k}_i.$$

Now, the next time step τ_{m+1} is proposed to be

$$\tau_{m+1} = \rho\frac{\tau_m^2}{\tau_{m-1}}\left(\frac{TOL \cdot r_m}{r_{m+1}^2}\right)^{1/p}, \quad (11)$$

where $\rho \in (0, 1]$ is a safety factor, $TOL > 0$ is a given tolerance and

$$r_{m+1} := \|\mathbf{u}_{m+1} - \hat{\mathbf{u}}_{m+1}\|. \quad (12)$$

This step size selection rule is called PI–controller [GLS88], and details on the numerical error and the implementation of the automatic steplength control can be found in [HW96, Lan01].

2.2 Rosenbrock–Wanner methods

Application to ODEs. As in the case of DIRK schemes we start our considerations with an implicit ODE of the form (3). A Rosenbrock–Wanner–method (ROW method) with s internal stages is given by

$$M\mathbf{k}_i = \mathbf{F}\left(t_m + \alpha_i\tau_m, \tilde{\mathbf{U}}_i\right) + \tau_m J \sum_{j=1}^{i} \gamma_{ij}\mathbf{k}_j + \tau_m\gamma_i\dot{\mathbf{F}}(t_m, \mathbf{u}_m), \tag{13}$$

$$\tilde{\mathbf{U}}_i = \mathbf{u}_m + \tau_m \sum_{j=1}^{i-1} a_{ij}\mathbf{k}_j, \quad i = 1, \ldots, s,$$

$$\mathbf{u}_{m+1} = \mathbf{u}_m + \tau_m \sum_{i=1}^{s} b_i\mathbf{k}_i, \tag{14}$$

where $J := \partial_{\mathbf{u}}\mathbf{F}(t_m, \mathbf{u}_m)$, α_{ij}, γ_{ij}, b_i are the parameters of the method,

$$\alpha_i := \sum_{j=1}^{i-1} \alpha_{ij}, \quad \gamma_i := \sum_{j=1}^{i-1} \gamma_{ij}, \quad \gamma := \gamma_{ii} > 0,\ i = 1, \ldots, s.$$

If the parameters α_{ij}, γ_{ij}, and b_i are chosen appropriately, a sufficient consistency order can be obtained. Additional consistency conditions arise if J is only an approximation to $\partial_{\mathbf{u}}\mathbf{F}(t_m, \mathbf{u}_m)$, or if J is an arbitrary matrix. This class of methods are called W–methods, [SW92]. If a ROW method is applied to semidiscretized partial differential equation, further order conditions should be satisfied to avoid order reduction, see [LO95]. The same stability concepts apply for ROW methods as for DIRK–methods.

The ROW method (13)–(14) requires the successive solution of s linear systems of equations with the same matrix $M - \gamma\tau_m J$. Note, J depends only on $\mathbf{u}_m$. The right hand side of the i–th linear system of equations depends on the solutions of the first to the $(i-1)$–st system. Thus, a main difference of ROW methods to DIRK methods is that it is not necessary to solve a nonlinear system of equations in each discrete time but a fixed number of linear systems of equations, i.e. there appears no iteration loop for solving the nonlinear system and so the method can be interpreted as iteration-free.

Again, as in the last section about diagonal–implicit RK–methods, an automatic step length control can be implemented with the help of an embedded method. Common Rosenbrock methods as ROS3P, ROS3Pw have an embedded method.

Application to strongly coupled systems. An ROW method applied on the strongly

coupled system (1)–(2) reads as

$$\begin{aligned}
&\begin{pmatrix} M_1 - \tau\gamma\partial_{\mathbf{u}}\mathbf{f} & -\tau\gamma\partial_{\mathbf{v}}\mathbf{f} \\ -\tau\gamma\partial_{\mathbf{u}}\mathbf{g} & M_2 - \tau\gamma\partial_{\mathbf{v}}\mathbf{g} \end{pmatrix} \begin{pmatrix} \mathbf{U}_i \\ \mathbf{V}_i \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{f}\left(t_m + \alpha_i\tau_m, \hat{\mathbf{U}}_i, \hat{\mathbf{V}}_i\right) \\ \mathbf{g}\left(t_m + \alpha_i\tau_m, \hat{\mathbf{U}}_i, \hat{\mathbf{V}}_i\right) \end{pmatrix} + \tau_m \sum_{j=1}^{i-1} \gamma_{ij} \begin{pmatrix} \partial_{\mathbf{u}}\mathbf{f} & \partial_{\mathbf{v}}\mathbf{f} \\ \partial_{\mathbf{u}}\mathbf{g} & \partial_{\mathbf{v}}\mathbf{g} \end{pmatrix} \begin{pmatrix} \mathbf{U}_i \\ \mathbf{V}_i \end{pmatrix} \\
&+ \tau_m\gamma_i \begin{pmatrix} \dot{\mathbf{f}}\left(t_m + \alpha_i\tau_m, \hat{\mathbf{U}}_i, \hat{\mathbf{V}}_i\right) \\ \dot{\mathbf{g}}\left(t_m + \alpha_i\tau_m, \hat{\mathbf{U}}_i, \hat{\mathbf{V}}_i\right) \end{pmatrix},
\end{aligned} \tag{15}$$

$$\hat{\mathbf{U}}_i = \mathbf{u}_m + \tau_m \sum_{j=1}^{i-1} a_{ij}\mathbf{U}_j, \quad \hat{\mathbf{V}}_i = \mathbf{u}_m + \tau_m \sum_{j=1}^{i-1} a_{ij}\mathbf{V}_j, \quad i = 1, \ldots, s, \tag{16}$$

$$\mathbf{u}_{m+1} = \mathbf{u}_m + \sum_{i=1}^{s} b_i\mathbf{U}_i, \mathbf{v}_{m+1} = \mathbf{v}_m + \sum_{i=1}^{s} b_i\mathbf{V}_i. \tag{17}$$

3 THE PARTITIONED APPROACH

Next we consider the partitioned approach, i.e. the ODEs (1)–(2) are solved with different codes and methods. Considering the DIRK– and ROW–methods the systems (10) and (15)–(16) are solved on different computers. Therefore we start our considerations with the strongly coupled, non-linear system

$$0 = \mathbf{f}(\mathbf{x}, \mathbf{y}), \qquad \mathbf{x} \in \mathbb{R}^m \tag{18}$$

$$0 = \mathbf{g}(\mathbf{x}, \mathbf{y}), \qquad \mathbf{y} \in \mathbb{R}^n, \tag{19}$$

where $\mathbf{f} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^m$ and $\mathbf{g} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$ are sufficiently smooth functions. We discuss two partitioned approaches to the solve the problem (18)–(19). First the Block–Gauß–Seidel method and second the Block–Newton method.

3.1 The Block–Gauß–Seidel method

The idea of the Block–Gauß–Seidel method is the following. First we solve equation (18) with fixed $\mathbf{y}$ w.r.t. $\mathbf{x}$. Then we insert this solution $\mathbf{x}$ into equation (19) and solve it w.r.t. $\mathbf{y}$. This procedure is repeated until convergence and a graphical illustration of the method can be found in Figure 1. For the convergence of the Block–Gauß–Seidel method we refer to [Axe96].

Application to DIRK–methods In this section we apply the Block–Gauß–Seidel method on our coupled system (6)–(7). As explained before we use a simplified Newton iteration to solve the two non-linear systems. Our parititioned method reads then as follows:

1. Set $\nu := 0$, $\mathbf{k}_i^{(\nu)} := 0$ and $\mathbf{l}_i^{(\nu)} := 0$.

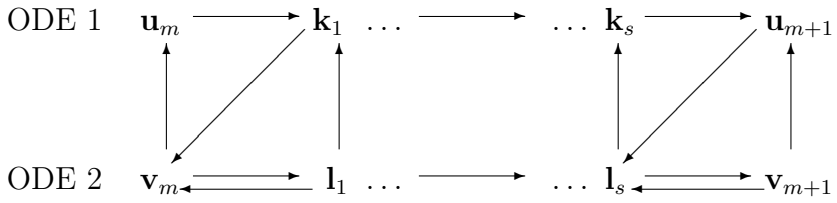

Figure 1: Block–Gauß–Seidel method

2. Compute
$$\mathbf{V}_i^{(\nu)} := \mathbf{v}_m + \tau_m \sum_{j=1}^{i-1} a_{ij}\mathbf{l}_j + \tau_m a_{ii}\mathbf{l}_i^{(\nu)}$$
and communicate it to the first solver.

3. Compute $\mathbf{k}_i^{(\nu+1)}$ by solving (6), set
$$\mathbf{U}_i^{(\nu+1)} := \mathbf{v}_m + \tau_m \sum_{j=1}^{i-1} a_{ij}\mathbf{k}_j + \tau_m a_{ii}\mathbf{k}_i^{(\nu+1)},$$
and communicate $\mathbf{U}_i^{(\nu+1)}$ to the second solver.

4. Compute $\mathbf{l}_i^{(\nu+1)}$ by solving (7).

5. Set $\nu := \nu + 1$.

6. If the values $\mathbf{k}_i^{(\nu+1)}$ and $\mathbf{l}_i^{(\nu+1)}$ are not sufficiently accurate then go to Step 2

Application to ROW–methods In this section we apply the Block–Gauß–Seidel method on our coupled system (15). In this case to a simplified, iteration free Block–Gauß method since the diagonal blocks of the Jacobian, i.e. J_{12} and J_{21}, are set two zero. To get a better approximation we can manipulate the time derivative of the right-hand side, but as we will see later in the section on the numerical example our coupled Rosenbrock methods have order reduction.

Our parititioned method then reads as follows. Compute $\mathbf{U}_i$ by solving

$$(M_1 - \gamma\tau_m\partial_{\mathbf{u}}\mathbf{f})\,\mathbf{U}_i = \mathbf{f}\left(t_m + \alpha_i\tau_m, \hat{\mathbf{U}}_i, \hat{\mathbf{V}}_i\right) + \tau_m \sum_{j=1}^{i-1} \gamma_{ij}\partial_{\mathbf{u}}\mathbf{f}\mathbf{U}_j + \gamma_i\tau_m\left(\dot{\mathbf{f}}(t_m, \mathbf{u}_m) + \partial_{\mathbf{v}}\mathbf{f}\mathbf{g}_m\right)$$

and

$$(M_2 - \gamma\tau_m\partial_{\mathbf{v}}\mathbf{g})\,\mathbf{V}_i = \mathbf{g}\left(t_m + \alpha_i\tau_m, \hat{\mathbf{U}}_i, \hat{\mathbf{V}}_i\right) + \tau_m \sum_{j=1}^{i-1} \gamma_{ij}\partial_{\mathbf{v}}\mathbf{g}\mathbf{V}_j + \gamma_i\tau_m\left(\dot{\mathbf{g}}(t_m, \mathbf{u}_m) + \partial_{\mathbf{u}}\mathbf{g}\mathbf{f}_m\right)$$

with $\mathbf{f}_m := \mathbf{f}(t_m, \mathbf{u}_m, \mathbf{v}_m)$, $\mathbf{g}_m := \mathbf{g}(t_m, \mathbf{u}_m, \mathbf{v}_m)$, $\hat{\mathbf{U}}_i$ and $\hat{\mathbf{V}}_i$, $i = 1, \ldots, s$, given by (16). The terms $\partial_{\mathbf{v}}\mathbf{f}\mathbf{g}_m$ and $\partial_{\mathbf{u}}\mathbf{g}\mathbf{f}_m$ result from the time derivative of $\mathbf{f}$ and $\mathbf{g}$.

3.2 The Block–Newton method

First we apply Newton's method on our nonlinear system (18) and (19). We obtain the linear system

$$\begin{pmatrix} \mathbf{f}_x & \mathbf{f}_y \\ \mathbf{g}_x & \mathbf{g}_y \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x}^{(k+1)} \\ \Delta\mathbf{y}^{(k+1)} \end{pmatrix} = - \begin{pmatrix} \mathbf{f}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \\ \mathbf{g}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \end{pmatrix}, \tag{20}$$

where $\mathbf{f}_x$, ... are the Jacobians and $\Delta\mathbf{x}^{(k+1)} := \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$. In the next step we apply one Gauß-step on the system (20), i.e. we resolve the first equation of (20) w.r.t. $\Delta\mathbf{x}^{(k+1)}$, i.e.

$$\Delta\mathbf{x}^{(k+1)} = -\mathbf{f}_x^{-1}(\mathbf{f}_y\Delta\mathbf{y}^{(k+1)} + \mathbf{f})$$

and insert this result into the second equation, i.e.

$$-\mathbf{g}_x\mathbf{f}_x^{-1}(\mathbf{f}_y\Delta\mathbf{y}^{(k+1)} + \mathbf{f}) + \mathbf{g}_y\Delta\mathbf{y}^{(k+1)} = -\mathbf{g}$$

or

$$(\mathbf{g}_y - \mathbf{g}_x\mathbf{f}_x^{-1}\mathbf{f}_y)\Delta\mathbf{y}^{(k+1)} = \mathbf{g}_x\mathbf{f}_x^{-1}\mathbf{f} - \mathbf{g}.$$

For abbreviation we set

$$S := \mathbf{g}_y - \mathbf{g}_x \underbrace{\mathbf{f}_x^{-1}\mathbf{f}_y}_{=:C}, \quad \mathbf{p} := \mathbf{f}_x^{-1}\mathbf{f}.$$

The matrix S is often called Schur complement. Now our Block Elimination Algorithm has the following form

1. Compute $\mathbf{p}$, i.e. solve $\mathbf{f}_x\mathbf{p} = \mathbf{f}$ for $\mathbf{p}$.
2. Compute $C = \mathbf{f}_x^{-1}\mathbf{f}_y$, i.e. solve the matrix equation $\mathbf{f}_x C = \mathbf{f}_y$ for C.
3. Compute the Schur complement $S = \mathbf{g}_y - \mathbf{g}_x C$.
4. Compute the modified right-hand side $\mathbf{g}_x\mathbf{p} - \mathbf{g} =: \tilde{\mathbf{g}}$.
5. Solve $S\Delta\mathbf{y} = \tilde{\mathbf{g}}$ for $\Delta\mathbf{y}$.
6. Compute $\Delta\mathbf{x} = -(\mathbf{p} + C\Delta\mathbf{y})$.

Application to DIRK–methods In this paragraph we apply our Block–Newton method on the non-linear equation (6)–(7). Therefore, as in the previous section, we first apply a Newton method on (6)–(7) leading to the linear system (10) which is then solved by a Block–Gauß algorithm. Our Block–Newton method reads then as follows

1. Compute $\mathbf{p}$, i.e. solve $(M_1 - \tau a_{ii}\partial_{\mathbf{u}}\mathbf{f}_m)\mathbf{p} = \mathbf{f}\left(t_m + c_i\tau_m, \mathbf{U}_i^{(\nu)}, \mathbf{V}_i^{(\nu)}\right) - M_1\mathbf{k}_i^{(\nu)}$.
2. Compute $C = -\tau a_{ii}(M_1 - \tau a_{ii}\partial_{\mathbf{u}}\mathbf{f}_m)^{-1}\partial_{\mathbf{v}}\mathbf{f}_m$, i.e. solve the matrix equation
$$(M_1 - \tau a_{ii}\partial_{\mathbf{u}}\mathbf{f}_m)C = -\tau a_{ii}\mathbf{f}_m\partial_{\mathbf{v}}\mathbf{f}_m$$
for C.
3. Compute the Schur complement $S = (M_2 - \tau a_{ii}\partial_{\mathbf{v}}\mathbf{g}_m) + \tau a_{ii}\partial_{\mathbf{u}}\mathbf{g}_m C$.
4. Compute the modified right-hand side $\tau a_{ii}\partial_{\mathbf{u}}\mathbf{g}_m\mathbf{p} - M_2\mathbf{l}_i^{(\nu)} + \mathbf{g} =: \tilde{\mathbf{g}}$.
5. Solve $S\Delta\mathbf{y} = \tilde{\mathbf{g}}$ for $\Delta\mathbf{y}$.
6. Compute $\Delta\mathbf{x} = \mathbf{p} - C\Delta\mathbf{y}$.

Application to ROW–methods In the case of ROW–methods the Block–Newton method simplifies to a Block–Gauß method applied on the linear system (15). The method reads as

1. Compute $\mathbf{p}$, i.e. solve $(M_1 - \tau a_{ii}\partial_{\mathbf{u}}\mathbf{f}_m)\mathbf{p} = \mathbf{f}\left(t_m + c_i\tau_m, \mathbf{U}_i^{(\nu)}, \mathbf{V}_i^{(\nu)}\right)$.
2. Compute $C = \tau a_{ii}(M_1 - \tau a_{ii}\partial_{\mathbf{u}}\mathbf{f}_m\partial_{\mathbf{u}}$, i.e. solve the matrix equation
$$(M_1 - \tau a_{ii}\partial_{\mathbf{u}}\mathbf{f}_m)C = \tau a_{ii}\mathbf{f}_m)^{-1}\partial_{\mathbf{u}}$$
for C.
3. Compute the Schur complement $S = (M_2 - \tau a_{ii}\partial_{\mathbf{v}}\mathbf{g}_m) - \tau a_{ii}\partial_{\mathbf{u}}\mathbf{g}_m C$.
4. Compute the modified right-hand side $\tau a_{ii}\partial_{\mathbf{u}}\mathbf{g}_m\mathbf{p} - \mathbf{g} =: \tilde{\mathbf{g}}$.
5. Solve $S\Delta\mathbf{y} = \tilde{\mathbf{g}}$ for $\Delta\mathbf{y}$.
6. Compute $\Delta\mathbf{x} = -(\mathbf{p} + C\Delta\mathbf{y})$.

4 NUMERICAL EXAMPLES

As an example we consider a simple predator-prey model which is given by

$$\begin{aligned} \dot{u} &= 10u(1-v), \quad & u(0) &= 3 \\ \dot{v} &= v(u-1), & v(0) &= 1. \end{aligned}$$

For determining the numerical solution we apply the trapezoidual rule, DIRK3 [Ran07], ROS3P [RA05], and ROS3Pw [RA05] with equidistant time steps $\tau = 1/(10 \cdot 2^N)$, $N = 0, 1, dots, 5$.

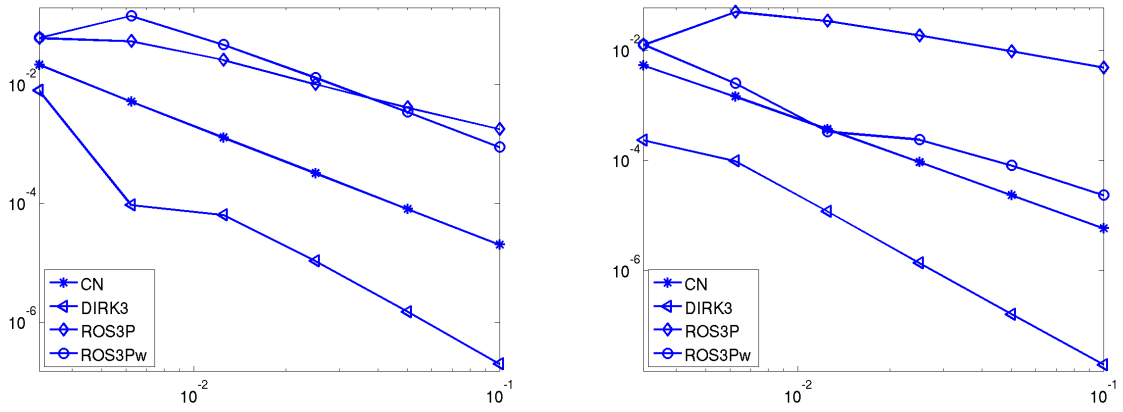

Figure 2: Block–Gauß–Seidel method: τ versus error

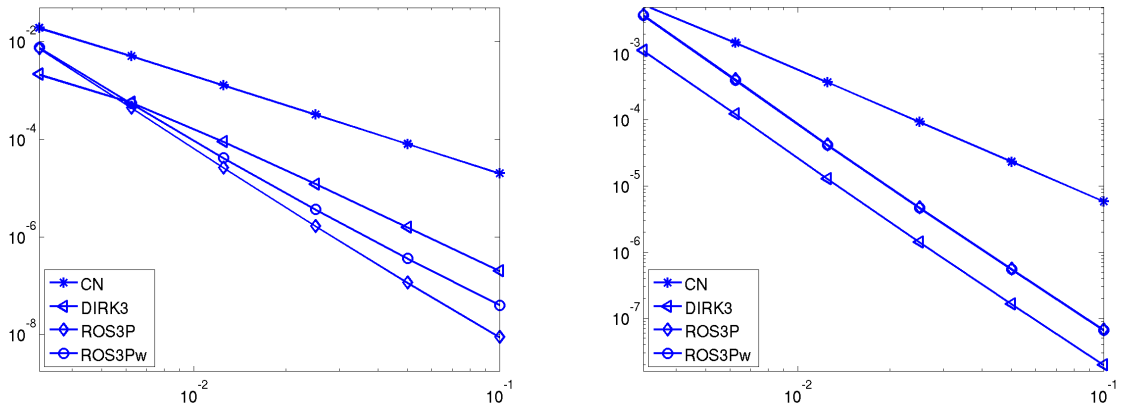

Figure 3: Block–Newton method: τ versus error

In Figure 2 we present the results which are obtained with the Block–Gauß–Seidel method. It can be observed that the ROW–methods have order reduction since the Jacobian is not evaluated exactly. The DIRK–methods give better results since these methods need not the evaluation of a Jacobian.

Different results appear if the Block–Newton method is used to solve the final system (see Figure 3). In this case all methods reach the desired order and the ROW–methods give much better results as then with the Block–Gauß–Seidel method.

5 SUMMARY AND OUTLOOK

In this note we have introduced the Block–Gauß–Seidel and the Block–Newton method for DIRK– and ROW–methods. In the case of ROW–methods we get an iteration-free, partitioned method.

In the next steps we first have to consider more complicated ODEs, e.g. the coupling of parabolic differential equations. Moreover the coupling of differential algebraic equations should be analysed since it is well-known that the Block–Gauß–Seidel method may fail for this class of problems.

REFERENCES

[Axe96] Owe Axelsson. *Iterative solution methods*. Cambridge Univ. Press., Cambridge, 1996.

[DR08] D. Dinkler and J. Rang, editors. *Wechselwirkung von Struktur und Fluid - Abschlussbericht eines Graduiertenkollegs*. Wolfram Schmidt Buchbinderei & Druckerei, Braunschweig, 2008.

[FP80] C.A. Felippa and K.C. Park. Staggered transient analysis procedures for coupled mechanical systems: Formulation. *Comput. Methods Appl. Mech. Eng.*, 24:61–111, 1980.

[GLS88] K. Gustafsson, M. Lundh, and G. Söderlind. A PI stepsize control for the numerical solution of ordinary differential equations. *BIT*, 28(2):270–287, 1988.

[GS00] P.M. Gresho and R.L. Sani. *Incompressible Flow and the Finite Element Method*. Wiley, Chichester, 2000.

[HH10] S. Hartmann and A.-W. Hamkar. Rosenbrock-type methods applied to finite element computations within finite strain viscoelasticity. *Computer Methods in Applied Mechanics and Engineering*, 199(23-24):1455–1470, 2010.

[HW96] E. Hairer and G. Wanner. *Solving ordinary differential equations. II: Stiff and differential-algebraic problems.*, volume 14 of *Springer Series in Computational Mathematics*. Springer, Berlin, 1996.

[JGR06] V. John, Matthies G., and J. Rang. A comparison of time-discretization/linearization approaches for the incompressible Navier–Stokes equations. *Comput. Methods Appl. Mech. Eng.*, 195:5995–6010, 2006.

[JR10] V. John and J. Rang. Adaptive time step control for the incompressible Navier–Stokes equations. *Comput. Methods Appl. Mech. Eng.*, 199:514–524, 2010.

[Lan01] J. Lang. *Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems*, volume 16 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin, 2001.

[LO95] C. Lubich and A. Ostermann. Linearly implicit time discretization of non-linear parabolic equations. *IMA J. Numer. Anal.*, 15(4):555–583, 1995.

[MM04] M. Meyer and H. G. Matthies. State-space representation of instationary two-dimensional airfoil aerodynamics. *Journal of Wind Engineering and Industrial Aerodynamics*, 92(3-4):263–274, 2004.

[MNS06] Hermann G. Matthies, Rainer Niekamp, and Jan Steindorf. Algorithms for strong coupling procedures. *Comput. Methods Appl. Mech. Eng.*, 195(17-18):2028–2049, 2006.

[MS02] H. G. Matthies and J. Steindorf. Partitioned but strongly coupled iteration schemes for nonlinear fluid-structure interaction. *Comput. Struct.*, 80:1991–1999, 2002.

[MS03] H. G. Matthies and J. Steindorf. Partitioned strong coupling algorithms for fluid-structure interaction. *Comput. Struct.*, 81:805–812, 2003.

[MW01] D. P. Mok and W. A. Wall. Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures. In K. Schweizerhof W. A. Wall, K.-U. Bletzinger, editor, *Trends in Computational Structural Mechanics*, pages 689–698. CIMNE, Barcelona, 2001.

[PFL95] S. Piperno, C. Farhat, and B. Larrouturou. Partitioned procedures for the transient solution of coupled aeroelastic problems. I: Model problem, theory and two-dimensional application. *Comput. Methods Appl. Mech. Eng.*, 124(1-2):79–112, 1995.

[RA05] J. Rang and L. Angermann. New Rosenbrock methods for partial differential algebraic equations of index 1. *BIT*, 45(4):761–787, 2005.

[Ran07] Joachim Rang. Design of DIRK schemes for solving the Navier-Stokes-equations. Informatik-Bericht 2007-02, TU Braunschweig, Braunschweig, 2007.

[RB00] S. Rugonyi and K.-J. Bathe. On the analysis of fully-coupled fluid flows with structural interactions — a coupling and condensation procedure. *Int. J. of Comp. Civil and Struct. Eng.*, 1:29–41, 2000.

[RSM09] Joachim Rang, Johannes Schöön, and Hermann G. Matthies. Solving FSI problems with high resolution and using a component framework in parallel. In Stefan Hartmann, Andreas Meister, Michael Schäfer, and Stefan Turek, editors, *International Workshop on Fluid-Structure Interaction: Theory, Numerics and Applications*, Kassel, 2009. Kassel Univ. Press.

[SW92] K Strehmel and R. Weiner. *Linear-implizite Runge–Kutta-Methoden und ihre Anwendung*, volume 127 of *Teubner-Texte zur Mathematik*. Teubner, Stuttgart, 1992.